Satisfiability of General Intruder Constraints with a Set Constructor

Tigran Avanesov
Loria-INRIA Grand Est
Nancy, France
Tigran.Avanesov@loria.fr

Yannick Chevalier
IRIT-Université Paul Sabatier
Toulouse, France
ychevali@irit.fr

Michaël Rusinowitch
Loria-INRIA Grand Est
Nancy, France
Michael.Rusinowitch@loria.fr

Mathieu Turuani
Loria-INRIA Grand Est
Nancy, France
Mathieu.Turuani@loria.fr

***Abstract*—Many decision problems on security protocols can be reduced to solving so-called intruder constraints in Dolev Yao model. Most constraint solving procedures for protocol security rely on two properties of constraint systems called *monotonicity* and *variable-origination*. In this work we relax these restrictions by giving an NP decision procedure for solving general intruder constraints (that do not have these properties). Our result extends a first work by L. Mazaré in several directions: we allow non-atomic keys, and an associative, commutative and idempotent symbol (for modeling sets). We also give several new applications of the result.**

***Keywords*-Security, constraint solving, Dolev-Yao intruder, general constraints, ACI.**

I. Introduction

Detecting flaws in security protocol specifications under the perfect cryptography assumption in Dolev-Yao intruder model is an approach that has been extensively investigated in recent years [1], [2], [3], [4]. In particular symbolic constraint solving has proved to be a very successful approach in the area. It amounts to express the possibility of mounting an attack, e.g. the derivation of a secret, as a list of steps where for each step some message has to be derived from the current intruder knowledge. These steps correspond in general to the progression of the protocol execution, up to the last one which is the secret derivation.

Enriching standard Dolev-Yao intruder model with different equational theories [5] like exclusive OR, modular exponentiation, Abelian groups, etc. [7], [8] helps to find flaws that could not be detected considering free symbols only. A particularly useful theory is the theory of an *ACI* operator (that is an associative commutative and idempotent one) since it allows one to express sets in cryptographic protocols.

Up to one exception [9], [10], all proposed algorithms rely on two strong assumptions about the constraints to be processed: knowledge monotonicity and variable origination. Constraints satisfying this hypothesis are called *well-formed constraints* in the literature and they are not restrictive as these conditions hold when handling standard security problems with a single Dolev-Yao intruder. However, we will see that in some situations it can be quite useful to relax these hypotheses and consider *general constraints*, that is constraints without the restrictions above. General constraints naturally occur when considering security problems involving several non-communicating Dolev-Yao intruders (see § II-A). Note that if intruders can communicate during protocol execution, the model becomes attack-equivalent to one with a unique Dolev-Yao intruder [11].

The work presented in this paper was partially supported by the FP7-ICT-2007-1 Project no. 216471, "AVANTSSAR: Automated Validation of Trust and Security of Service-oriented Architectures" (http://www.avantssar.eu)

A. Contributions of the paper

First, we will show that as for the standard case, in this more general framework it is still possible to derive an NP decision procedure for detecting attacks on a bounded number of protocol sessions (Sections V). Second, our result extends previous ones by allowing non-atomic keys and the usage of an associative commutative idempotent operator (Sections III, IV) that can be used for instance to model sets of nodes in an XML document (see § II-C). Finally we will sketch two applications of our results: colluding intruders and XML rewriting attacks (Section II).

B. Related works

The decision procedure for satisfiability of well-formed intruder constraint systems can be used to decide the insecurity of cryptographic protocols with a bounded number of sessions [13]. In this domain, several works deviated from the perfect cryptography assumption and considered algebraic properties of function symbols. For example properties of XOR operator and exponentiation were considered in [15], [16].

All the works mentioned above consider systems of constraints with two restrictions namely knowledge monotonicity (the left-hand side of a constraint is included into the left-hand side of the next one) and variable origination (variable appears first in right-hand side of some constraint): this limitation is not impeding the solution of usual protocol insecurity problems since the constraints generated with an active Dolev-Yao intruder are of the required type. An attempt to swerve from well-formed constraints was made by Mazaré [9]. He considered "quasi well-formed" constraint systems by relaxing the knowledge monotonicity. Later, in his thesis [10], he succeeded to find a decision procedure for satisfiability of general constraint systems with the restriction that keys

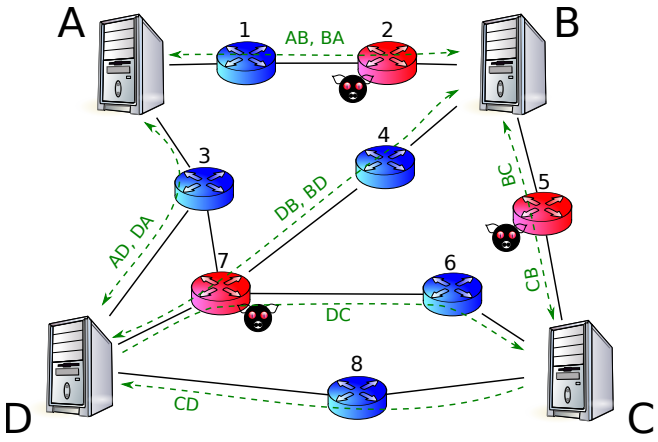

Figure 1. Untrusted routers

used for encryption are atomic. However to our knowledge no extension of Dolev-Yao deduction system to non-atomic keys or to algebraic properties has been shown decidable for general constraint systems. Moreover, satisfiability of well-formed constraints with ACI theory was not considered before.

II. Motivating examples

A. Protocol analysis with several intruders

In the domain of security protocol analysis Dolev-Yao model is widely used in spite of its limitations. We propose here to consider instead of a powerful Dolev-Yao intruder that controls the whole network, several non communicating Dolev-Yao intruders with smaller controlled domains. We give below an application of this model.

Suppose several agents ($A, B \ldots$, see Figure 1) execute a message exchange protocol (every agent has a finite list of actions in a send/receive format that is known to everybody). Due to their (long distance) layout they have to transmit data through routers ($1, 2, 3 \ldots$). The routing tables of all honest routers/agents are static (messages follow always the same path). Some routers ($2, 5, 7$) may be compromised: an intruder managed to install a device controlling input and output of the router or implanted there his malicious code. A message circulated via such an untrusted channel (e.g. DB) is consumed by the corresponding compromised device (*local intruder*) (7) thereby increasing his knowledge. Moreover, a local intruder can forge and emit to an endpoint (C, B, D) of any channel he controls (BD, DB, DC) any message he can build using the content of his memory and some available transformations specified by a deduction system. Because of the network topology malicious routers have no means to communicate (there is no links between them, neither direct nor via other routers), but at some point the intruder can gather the knowledge of all the compromised routers (by physically collecting devices or reading their memory).

In this framework the *security problem* is to know whether it is possible to initially give instructions to compromised routers to force such an execution that honest agents (that strictly follow their list of actions) will reveal some secret data to the intruder (i.e. intruder can build this data from the gathered at the end knowledge of all local intruders).

1) Formalizing the coordinated attack problem: To formalize the problem we introduce some notations and definitions. Let $\mathcal{A}$ be the set of atoms representing elementary pieces of data: texts, public keys, names of agents, etc. Let $\mathcal{X}$ be the set of variables, representing data holders for (possibly composed) values to be found. We define a term as follows (classically for protocol security domain, we allow encryption, pairing, etc.):

$$\begin{aligned} term \quad &::= variable \mid atom \mid \mathrm{pair}\,(term, term) \mid \\ &\quad \mathrm{enc}\,(term, term) \mid \cdot(tlist) \mid \mathrm{priv}\,(Keys) \mid \\ &\quad \mathrm{aenc}\,(term, Keys) \mid \mathrm{sig}\,(term, \mathrm{priv}\,(Keys)) \\ tlist \quad &::= term \mid term,\ tlist \end{aligned}$$

where $atom \in \mathcal{A}$, $variable \in \mathcal{X}$; $Keys \in \mathcal{A} \cup \mathcal{X}$. As we see, for asymmetric encryption (aenc) only atomic keys are allowed. By $\mathrm{sig}\,(p, \mathrm{priv}\,(a))$ we mean a signature of message p with private key $\mathrm{priv}\,(a)$; p is not deducible from the signature. Let $\mathcal{T}$ be the set of all possible terms. Let t be a term. We define $\sqrt{t}$ as a root symbol of t and $\mathrm{Vars}\,(t)$ the set of all variables in t. We call term t a *ground term*, if $\mathrm{Vars}\,(t) = \emptyset$. The set of ground terms is denoted by $\mathcal{T}_g$. We model sets using "$\cdot$" — a commutative, associative and idempotent (ACI) operator. To simplify the reasoning with ACI properties we define for every term t a unique normal form denoted by $\ulcorner t \urcorner$. It is defined by a strict total order $\prec$ on $\mathcal{T}$ and a normalization function that works bottom-up by flattening nested $\cdot$ lists ($\cdot(a, \cdot(c, d, e), c)$ becomes $\cdot(a, c, d, e, c)$), sorting children of $\cdot$-nodes and removing duplicates ($\cdot(a, c, d, e, c)$ becomes $\cdot(a, c, d, e)$). When the set is reduced to a singleton the ACI symbol is removed ($\cdot(a)$ becomes a). A term t is *normalized* if $t = \ulcorner t \urcorner$.

Example 1. *For a (non-normalized) term* t, $t = \cdot(a, \cdot(b, a, \mathrm{pair}\,(a, b)), \mathrm{pair}\,(\cdot(b, b), a))$ *we have* $\ulcorner t \urcorner = \cdot(\{a, b, \mathrm{pair}\,(a, b), \mathrm{pair}\,(b, a)\})$.

We will often implicitly extend a notation defined on terms to set of terms (also to list of terms and constraint systems) in a natural way, e.g. for a set of terms T we define $\ulcorner T \urcorner = \{\ulcorner t \urcorner : t \in T\}$, and for a constraint (defined later) $E \rhd t$, $\ulcorner E \rhd t \urcorner = \ulcorner E \urcorner \rhd \ulcorner t \urcorner$.

We define a substitution $\sigma = \{x_1 \mapsto t_1, \ldots, x_k \mapsto t_k\}$ (where $x_i \in \mathcal{X}$ and $t_i \in \mathcal{T}$) to be the mapping $\sigma : \mathcal{T} \to \mathcal{T}$, such that $t\sigma$ is a term obtained by replacing, for all i, each occurrence of variable x_i by the corresponding term t_i. Note that we are not allowed to apply $\sigma = \{x \mapsto \mathrm{pair}\,(a, b)\}$ to the term $\mathrm{aenc}\,(a, x)$, as the result is not a term. The set of variables $\{x_1, \ldots, x_k\}$ is called the *domain* of σ and

Table I
DY+ACI DEDUCTION SYSTEM RULES

Composition rules	Decomposition rules
$p, q \rightarrow \text{enc}(p, q)$	$\text{enc}(p, q), q \rightarrow p$
$p, q \rightarrow \text{aenc}(p, q)$	$\text{aenc}(p, q), \text{priv}(q) \rightarrow p$
$p, q \rightarrow \text{pair}(p, q)$	$\text{pair}(p, q) \rightarrow p$
$p, \text{priv}(q) \rightarrow \text{sig}(p, \text{priv}(q))$	$\text{pair}(p, q) \rightarrow q$
$t_1, \ldots, t_m \rightarrow \ulcorner \cdot(t_1, \ldots, t_m) \urcorner$	$\cdot(t_1, \ldots, t_m) \rightarrow t_i$, for all i

denoted by $\text{dom}(\sigma)$. A substitution σ is *ground* if for any $i \in \{1, \ldots, k\}$, t_i is ground; it is is normalized, iff $\forall x \in \text{dom}(\sigma)$, $x\sigma$ is normalized.

Agents: We will call communicating parties *agents*. Every agent is identified by its name.

Channels: From any agent a to any agent b there exists a directed communication channel which we denote as $a \rightharpoonup b$. The set of channels is denoted as $\mathbb{C}$. Each channel has a queue for storing incoming messages before they are processed.

Agents behavior: Every agent a has a finite list of actions l^a to execute in sequence. These are of two kinds:

- $?_f r$: agent a accepts message m matching term r modulo ACI ($\ulcorner r\sigma \urcorner = \ulcorner m \urcorner$ for some ground σ) admittedly from agent f on channel $f \rightharpoonup a$, instantiates $\text{Vars}(r)$ with σ and executes its remaining actions using these values. A reception of an incorrect message forces a to quit the protocol. Note that no notification is sent to the sender, thus a sender continues his execution[1].
- $!_t s$: agent a sends s to agent t over the channel $a \rightharpoonup t$.

Intruder model: We assume that some communication channels are controlled by *local* intruders and no channel is controlled by more than one intruder.

Let $\mathbb{I} = \{I_i\}_{i=1,\ldots,N}$ be the set of local intruders. We introduce an *intruder layout* represented by a function $\iota : \mathbb{C} \mapsto \mathbb{I} \cup \{\varnothing\}$, mapping every channel to the intruder that controls it if there is one or $\varnothing$ otherwise.

Every local intruder $I \in \mathbb{I}$ has some initial knowledge K_I^0 that is a set of ground terms. Intruder I eavesdrops and blocks all the messages over channels $\{c : \iota(c) = I\}$. He can also impersonate agents and send messages over these channels.

The set of intruder's deduction rules are defined on normalized terms and shown in Table I.

Term t is derivable from a set of terms E iff $t \in E$ or there exists a sequence of sets of ground terms $E_0, E_1, \ldots, E_n$ such that $E_0 = E$, $E_n \setminus E_{n-1} = \{t\}$ and $E_0 \rightarrow E_1 \rightarrow \cdots \rightarrow E_n$, where $A \rightarrow B$ is true iff there exists a (in our case DY+ACI) rule $L \rightarrow r$ and a substitution σ such that $L\sigma \subseteq A$ and $B \setminus A = \{r\sigma\}$. This sequence $E_0, E_1, \ldots, E_n$ is called a *derivation*. Note that $\text{Der}(\ulcorner E \urcorner)$ is normalized.

In our terminology, a local intruder I can send a message m on channel c, if $\iota(c) = I$ and $\ulcorner m \urcorner \in \text{Der}(\ulcorner K_I \urcorner)$, where K_I is his current knowledge.

Offline communication: At some point, the current knowledge of all local intruders can be gathered to derive a secret, which, probably, separately they cannot deduce. To do this, intruders must quit the protocol first.

Protocol scenario: We will call a *protocol scenario* a list of actions obtained by interleaving the action lists of all agents.

Coordinated attack problem: Input: Given a finite set of agents A together with their finite lists of actions $\{l^a\}_{a \in A}$ (we assume $\text{Vars}(l^a) \cap \text{Vars}(l^b) = \emptyset$, if $a \neq b$), a set of intruders $\{I_i\}_{i=1,\ldots,N}$ together with their initial knowledge K_{I_i}, an intruder layout mapping ι and a sensitive data represented by a ground term s_0.
Output: An executable instance of a protocol scenario such that s_0 can be derived from the union of local intruder knowledge collected after this execution (if one exists).

2) Solving the problem: We consider every protocol scenario $\{\langle a_i, \rho_i \rangle\}_{i=1,\ldots,k}$ of all possible lengths k, where ρ_i is an action from l^{a_i} (note that there are finitely many ones). According to the protocol scenario, we express its executability by a set of constraints (one for each message reception).

Definition II.1. *Let E be a set of terms and t be a term, we define the couple (E, t) denoted by $E \triangleright t$ to be a* constraint. *A* constraint system *is a finite set of constraints $\mathcal{S} = \{E_i \triangleright t_i\}_{i=1,\ldots,n}$.*

Definition II.2. *A ground substitution σ is a* model *of the constraint $E \triangleright t$ (or σ satisfies the constraint), if $\ulcorner t\sigma \urcorner \in \text{Der}(\ulcorner E\sigma \urcorner)$. It is a* model *of a constraint system $\mathcal{S}$, if it satisfies all the constraints of $\mathcal{S}$ and $\text{dom}(\sigma) = \text{Vars}(\mathcal{S})$.*

Example 2. *We give an example of a general constraint system and its solution within DY+ACI deduction system.*

$$\mathcal{S} = \left\{ \begin{array}{lcl} \text{enc}(x, a), \text{pair}(c, a) & \triangleright & b \\ \cdot(\{x, c\}) & \triangleright & a \end{array} \right\},$$

where $a, b, c \in \mathcal{A}$ and $x \in \mathcal{X}$.

One of the models is $\sigma = \{x \mapsto \text{enc}(\text{pair}(a, b), c)\}$.

The following lemma shows how can we encode unifiability of two terms modulo ACI by a deducibility constraint.

Lemma 1. *Substitution σ satisfies $p =_{ACI} q$ (i.e. $\ulcorner p\sigma \urcorner = \ulcorner q\sigma \urcorner$) iff it satisfies $\{\text{enc}(p, p)\} \triangleright \text{enc}(q, q)$.*

Using this encoding we can express the executability of a protocol scenario purely with a set of deducibility constraints avoiding unifications.

Going back to the coordinated attack problem, we build step by step a constraint system for expressing the scenario executability. Let $\mathcal{S} = \emptyset$. Running i from 1 to k we have:

- if $\rho_i = !_t s$ and $\iota(a_i \rightharpoonup t) = I$, then add s to K_I;
  but if $\iota(a_i \rightharpoonup t) = \varnothing$, then add s to queue of $a_i \rightharpoonup t$.

[1]One of the way to model another behavior, is to explicitly provide for every sending a succedent receive of an acknowledge message and for every receive a succedent send of an acknowledge message.

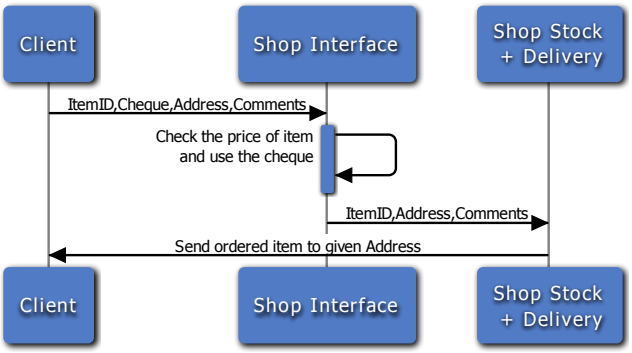

Figure 2. Ordering item scenario

- if $\rho_i = ?_f r$ and $\iota(f \rightharpoonup a_i) = I$, then add $K_I \rhd r$ to $\mathcal{S}$; but if $\iota(f \rightharpoonup a_i) = \varnothing$, then fail if queue of $f \rightharpoonup a_i$ is empty, otherwise let s be the first term in this queue: we add $\{\mathrm{enc}(s, s)\} \rhd \mathrm{enc}(r, r)$ to $\mathcal{S}$.

Finally, we add to $\mathcal{S}$ constraint $\bigcup_{I \in \mathbb{I}} K_I \rhd s_0$ expressing that s_0 can be obtained from the conjoint knowledge of all local intruders. Note that the resulting constraint system $\mathcal{S}$ is not mandatory well-formed. Now, if σ satisfies $\mathcal{S}$, then we output a considered scenario instantiated with σ.

In this way we reduced coordinated attack problem to solving constraint systems.

B. Attacks exploiting XML format

Here we show how to model (using our formalism) attacks based on an XML-representation of messages. A different technique to handle this kind of attacks was presented in [19].

We consider an e-shop that accepts e-cheques, and we suppose that it is presented by a Web Service using SOAP protocol for exchanging messages.

It consists of two services:

- the first exposes the list of goods for sale with their prices and process the orders by accepting payment,
- the second is a delivery service; it receives information from the first one about successfully paid orders, and sends the ordered goods to the buyer.

A simple scenario for ordering item is shown in Figure 3. First, a client sends an order using e-shop interface that consists of an item identifier, e-cheque, delivery address and some comments. Then, the first service of the e-shop checks whether the price of the ordered item corresponds to the received cheque. If it does, the service consumes the cheque and resends the order to the stock/delivery service (without the used e-cheque). Stock and delivery service prepare a parcel with ordered item and send it to given address. The comment is automatically printed on the parcel to give some complementary information about the delivery.

Suppose, Alice has an e-cheque for 5€. She selected a simple pen (with ItemID $simple$) to buy, but she liked very much a more expensive gilded one (ItemID $gilded$). Can we help Alice to get what she wants for what she has?

Let us formalize the behavior of scenario players (terms, normalization function and deduction system are defined as in § III-A1 except that we will write $(t_1 \cdot \ldots \cdot t_n)$ instead of $\cdot(t_1, \ldots, t_n)$). Identifiers starting from a capital letter are considered as variables; numbers and identifier starting from lower-case letter are considered as constants. We model a delivery of an item with some $ItemID$ to address $Addr$ with comments $Comm$ by the following message: $\mathrm{sig}((ItemID \cdot Addr \cdot Comm), \mathrm{priv}(k_s))$ — a message signed by e-shop, where k_s its public key such that no one can produce this message except the shop. We abstract away from the procedure of checking price of the item and will suppose that Shop Interface expects 5€ e-cheque for Item "$simple$". For simplicity we assume only two items.

We will use notation for sending and receiving as in § II-A.

For Shop Interface we have:

$$?_{Client}(simple \cdot cheque5 \cdot IAddr \cdot IComm);$$
$$!_{Delivery}(simple \cdot IAddr \cdot IComm).$$

For Shop Stock/Delivery we have:

$$?_{Interface}(DItemID \cdot DAddr \cdot DComm);$$
$$!_{Client}\,\mathrm{sig}((DItemID \cdot DAddr \cdot DComm), \mathrm{priv}(k_s)).$$

Alice initially has:

$simple, gilded$:	items identifiers;
$cheque5$:	an e-cheque for 5€;
$addr$:	her address;
$cmnts$:	residence digital code;
k_s:	a public key of the shop.

Now we build a mixed constraint system (derivation constraints and equations) to know, whether Alice can do what she wants (see Figure 4).

Constraint (1) shows that Alice can construct a message expected by the shop from a client. Constraint (2) represents a request from the first to the second service of the shop: left-hand side is a message sent by the interface service, and right-hand side is a message expected by stock/delivery service. The last constraint shows that from the received values Alice can build a message that models a delivery of item with ItemID gilded.

Then, one of the solutions is:

$$\begin{array}{llll} IAddr & \mapsto addr & IComm & \mapsto (gilded \cdot cmnts) \\ DItemID & \mapsto gilded & DAddr & \mapsto addr \\ & & DComm & \mapsto (simple \cdot cmnts) \end{array}$$

From this solution we see that Alice can send a not well-formed comments (that presents two XML-nodes), and Delivery service parser can choose an entry with ID gilded, while a parser of the first layer can return a value of the first occurrence of ItemID, i.e. simple. An attack-request can look like this:

$$\left\{\begin{array}{lll} \{gilded, simple, cheque5, addr, cmnts, k_s\} & \rhd & (simple \cdot cheque5 \cdot IAddr \cdot IComm) \\ (simple \cdot IAddr \cdot IComm) & =_{ACI} & (DItemID \cdot DAddr \cdot DComm) \\ \{gilded, simple, cheque5, addr, cmnts, k_s, & & \\ \mathrm{sig}\left((DItemID \cdot DAddr \cdot DComm), \mathrm{priv}\left(k_s\right)\right)\} & \rhd & \mathrm{sig}\left((gilded \cdot addr \cdot DComm), \mathrm{priv}\left(k_s\right)\right) \end{array}\right. \begin{array}{l} (1) \\ (2) \\ \\ (3) \end{array}$$

Figure 3. A constraint system describing possible vulnerability in E-shop scenario

```
<ItemID>simple</ItemID>
<Cheque>cheque5</Cheque>
<Address>addr</Address>
<Comments>cmnts</Comments>
<ItemID>gilded</ItemID>
```

This attack is possible, if Alice constructs a request "by hand", but a similar attack is probably feasible using XML-injection: Alice when filling a request form enters instead of her comments the following string:

```
cmnts</Comments>
<ItemID>gilded</ItemID><Comments>
```

and in the resulting request we get:

```
<ItemID>simple</ItemID>
<Cheque>cheque5</Cheque>
<Address>addr</Address>
<Comments>cmnts</Comments>
<ItemID>gilded</ItemID><Comments>
</Comments>
```

This kind of XML-injection attacks was described in [20].

III. Satisfiability of general DY+ACI constraints

In Section II we reduced several problems to solving deducibility constraints. In this section we present a decision procedure for a constraint system where Dolev-Yao deduction system is extended by an associative-commutative-idempotent symbol (DY+ACI). We consider, as before, operators for pairing, symmetric and asymmetric encryptions, decryption, signature and an ACI operator used as a set constructor.

After introducing necessary notations and stating auxiliary properties, we show that it suffices to consider normalized constraint systems and normalized models. Then we prove the existence of a conservative solution of satisfiable constraint system: it can be built using only quasi-subterms and priv-ed atoms of the constraint system. At the end we give a bound on the size of such a solution. Full proofs are given in [21].

A. Definitions and formal introduction to the problem

1) The notions: We will use "bin" to denote any binary operators: $\mathrm{bin} \in \{\mathrm{enc}, \mathrm{aenc}, \mathrm{pair}, \mathrm{sig}\}$. The cardinality of a set P is denoted by $|P|$.

Definition III.1. *For any term $t \in \mathcal{T}$ we define its* set of elements *by:*

$$\mathrm{elems}\left(t\right) = \begin{cases} \bigcup_{p \in L} \mathrm{elems}\left(p\right) & \text{if } t = \cdot\left(L\right); \\ \{t\}, & \text{otherwise.} \end{cases}$$

Example 3. *Referring to Example 1, we have* $\mathrm{elems}\left(t\right) = \{a, b, \mathrm{pair}\left(\cdot\left(b, b\right), a\right), \mathrm{pair}\left(a, b\right)\}$.

We define $\mathrm{Sub}(t)$ as the set of subterms of t.

Definition III.2. *Let t be a term. We define a set of quasi-subterms* $\mathrm{QSub}\left(t\right)$ *as follows:* $\mathrm{QSub}\left(t\right) =$

$$= \begin{cases} \{t\}, & \text{if } t \in \mathcal{X} \cup \mathcal{A}; \\ \{t\} \cup \mathrm{QSub}\left(t_1\right), & \text{if } t = \mathrm{priv}\left(t_1\right); \\ \{t\} \cup \mathrm{QSub}\left(t_1\right) \cup \mathrm{QSub}\left(t_2\right), & \text{if } t = \mathrm{bin}\left(t_1, t_2\right); \\ \{t\} \cup \bigcup_{p \in \mathrm{elems}(L)} \mathrm{QSub}\left(p\right), & \text{if } t = \cdot\left(L\right) \end{cases}$$

We denote $\mathrm{QSub}\left(\mathcal{S}\right) \setminus \mathcal{X}$ as $\mathrm{QS\mathring{u}b}\left(S\right)$.

Example 4. *Referring to Example 1, we have* $\mathrm{QSub}\left(t\right) = \{t, a, b, \mathrm{pair}\left(a, b\right), \mathrm{pair}\left(\cdot\left(b, b\right), a\right), \cdot\left(b, b\right)\}$ *and* $\mathrm{Sub}\left(t\right) = \mathrm{QSub}\left(t\right) \cup \{\cdot\left(b, a, \mathrm{pair}\left(a, b\right)\right)\}$.

Note that for any normalized term t, $\mathrm{QSub}\left(t\right) = \mathrm{Sub}\left(t\right)$.

We define a DAG-size of a term as the number of its different subterms (a size of the natural representation of a term).

Definition III.3. *We define a DAG-size of a term t as* $\mathrm{size}\left(t\right) = |\mathrm{Sub}\left(t\right)|$. *For $T \subseteq \mathcal{T}$,* $\mathrm{size}\left(T\right) = |\mathrm{Sub}\left(T\right)|$.

Definition III.4. *Let $T = \{t_1, \ldots, t_k\}$ be a non-empty set of terms. Then we define $\pi(T)$ as follows:* $\pi(T) = \ulcorner\cdot\left(t_1, \ldots, t_k\right)\urcorner$.

We consider hereinafter only constraint systems $\mathcal{S}$ such that $\mathrm{QSub}\left(\mathcal{S}\right) \cap \mathcal{A} \neq \emptyset$. Now we introduce a transformation $\pi(H^{\mathcal{S},\sigma}\left(\cdot\right))$ on ground terms that replaces recursively all binary functional symbols such that they are different from all the non-variable quasi-subterms of the constraint system instantiated with σ, with ACI symbol $\cdot$. Later, we will show that whenever σ is a model of $\mathcal{S}$ then so is $\pi(\mathrm{H}\left(\sigma\right))$.

Definition III.5. *Given a constraint system $\mathcal{S}$ and its model σ. Let us fix some $\alpha \in (\mathcal{A} \cap \mathrm{QSub}\left(\mathcal{S}\right))$. We define a function*

$H^{\mathcal{S},\sigma}(\cdot) : \mathcal{T}_g \to 2^{\mathcal{T}_g}$ *as follows:* $H^{\mathcal{S},\sigma}(t) =$

$$= \begin{cases} \{\alpha\}, & \text{if } t \in (\mathcal{A} \setminus \mathrm{QSub}(\mathcal{S})); \\ \{t\}, & \text{if } t \in (\mathcal{A} \cap \mathrm{QSub}(\mathcal{S})); \\ \{\mathrm{priv}\left(\pi(H^{\mathcal{S},\sigma}(t_1))\right)\}, & \text{if } t = \mathrm{priv}(t_1); \\ \{\mathrm{bin}(\pi(H^{\mathcal{S},\sigma}(t_1)), \pi(H^{\mathcal{S},\sigma}(t_2)))\}, & \text{if } t = \mathrm{bin}(t_1,t_2) \wedge \ulcorner t \urcorner \in \ulcorner \mathrm{QSub}(S)\sigma \urcorner; \\ H^{\mathcal{S},\sigma}(t_1) \cup H^{\mathcal{S},\sigma}(t_2), & \text{if } t = \mathrm{bin}(t_1,t_2) \wedge \ulcorner t \urcorner \notin \ulcorner \mathrm{QSub}(S)\sigma \urcorner; \\ \bigcup_{p \in L} H^{\mathcal{S},\sigma}(p), & \text{if } t = \cdot(L). \end{cases}$$

Henceforward, we will omit parameters and write $H(\cdot)$ instead of $H^{\mathcal{S},\sigma}(\cdot)$ for shorter notation.

Definition III.6. *Let* $\theta = \{x_1 \mapsto t_1, \ldots, x_k \mapsto t_k\}$ *be a substitution. We define* $\pi(H(\theta))$ *the substitution* $\{x_1 \mapsto \pi(H(t_1)), \ldots, x_k \mapsto \pi(H(t_k))\}$.

Example 5. *We refer to Example 2 and show that* $\pi(H(\sigma))$ *is also a model of* $\mathcal{S}$. $\pi(H(\mathrm{enc}(\mathrm{pair}(a,b),c))) = \pi(H(\mathrm{pair}(a,b)) \cup \{c\}) = \pi(\{a\} \cup \{b\} \cup \{c\}) = \cdot(a,b,c)$ *(we suppose that* $a \prec b \prec c$*). One can see that* $\pi(\mathrm{H}(\sigma)) = \{x \mapsto \cdot(a,b,c)\}$ *is also a model of* $\mathcal{S}$ *within DY+ACI.*

2) General properties used in the proof: In Lemma 2 we list some of the auxiliary properties that can be used in the proof.

Lemma 2. *The following statements are true:*

1) $\ulcorner \mathrm{elems}(t) \urcorner = \mathrm{elems}(\ulcorner t \urcorner)$
2) $\pi(T) = \pi(\ulcorner T \urcorner)$
3) $H(t) = H(\ulcorner t \urcorner)$
4) $\pi(T_1 \cup \cdots \cup T_m) = \pi(\{\pi(T_1), \ldots, \pi(T_m)\})$
5) $\mathrm{QSub}(\ulcorner t \urcorner) \subseteq \ulcorner \mathrm{QSub}(t) \urcorner$
6) $\mathrm{QSub}(t\sigma) \subseteq \mathrm{QSub}(t)\sigma \cup \mathrm{QSub}(\mathrm{Vars}(t)\sigma)$
7) $\forall s \in \mathrm{Sub}(t)\ \ \mathrm{size}(\ulcorner t\sigma \urcorner) \geq \mathrm{size}(\ulcorner s\sigma \urcorner)$.

In Proposition 1 we remark that ACI-set of normalized terms has the same deductive expressiveness as that set of normalized terms itself.

Proposition 1. *Let* T *be a set of terms* $T = \{t_1, \ldots, t_k\}$. *Then* $\pi(T) \in \mathrm{Der}(\ulcorner T \urcorner)$ *and* $\ulcorner T \urcorner \subseteq \mathrm{Der}(\{\pi(T)\})$.

The possibility of working only with normalized versions of constraint systems and models justified in Proposition 2.

Proposition 2. σ *is a model of* $\mathcal{S}$ *iff* σ *is a model of* $\ulcorner \mathcal{S} \urcorner$ *iff* $\ulcorner \sigma \urcorner$ *is a model of* $\mathcal{S}$.

B. Existence of conservative solutions

In this subsection we will show that for any satisfiable constraint system, there exist a model in special form called *conservative* solution. Roughly speaking, a model in this form can be defined per each variable by set of quasi-subterms and atoms (that must be "priv"-ed) of the constraint system. This will bound a search space for the model (see § III-C).

First, we show that on quasi-subterms of constraint system instantiated with its model, the transformation $\pi(H(\cdot))$ will be a homomorphism modulo normalization.

Proposition 3. *Given a normalized constraint system* $\mathcal{S}$ *and its normalized model* σ. *For all* $t \in \mathrm{QSub}(\mathcal{S})$, $\ulcorner t\,\pi(\mathrm{H}(\sigma)) \urcorner = \ulcorner \pi(H(t\sigma)) \urcorner$.

Proof idea: Use induction on $\mathrm{size}(t)$. ■

Now we show that relation of derivability between a term and a set of terms is stable with regard to transformation $\pi(H(\cdot))$.

Lemma 3. *Given a normalized constraint system* $\mathcal{S}$ *and its normalized model* σ. *For any ground instance of DY+ACI rule* $l_1, \ldots, l_k \to r$, $\pi(H(r)) \in \mathrm{Der}(\{\pi(H(l_1)), \ldots, \pi(H(l_k))\})$.

Proof idea: Proof can be done by considering all DY+ACI rules. For cases where bin *appears in a rule (e.g.* $p, q \to \mathrm{enc}(p,q)$*), one should consider two possibilities:* $\exists u \in \mathrm{QSub}(S)$ *such that* $\ulcorner \mathrm{bin}(p,q) \urcorner = \ulcorner u\sigma \urcorner$ *or not. Proposition 1 is very useful for the proof.* ■

Using Proposition 3 and Lemma 3 we can prove the theorem below by simply following the transformed with $\pi(H(\cdot))$ derivation that proves σ is a model of $\mathcal{S}$.

Theorem 1. *Given a normalized constraint system* $\mathcal{S}$ *and its normalized model* σ. *Then* $\pi(\mathrm{H}(\sigma))$ *also satisfies* $\mathcal{S}$.

Now we define a conservative solution.

Definition III.7. *A normalized model* σ *of constraint system* $\mathcal{S}$ *is called* conservative, *iff for all* $x \in \mathrm{dom}(\sigma)$ *there exist different* $s_1, \ldots, s_k \in \mathrm{QSub}(S) \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})$ *such that* $\forall i \sqrt{s_i} \neq$ "$\cdot$" *and* $x\sigma = \pi(\{s_1\sigma, \ldots, s_k\sigma\})$.

Next, we show an existence of a conservative solution for a satisfiable constraint system $\mathcal{S}$, if its normalized model σ sends different variables to different values.

Proposition 4. *Given a normalized constraint system* $\mathcal{S}$ *and its normalized model* σ *such that* $\forall x, y \in \mathrm{dom}(\sigma), x \neq y$ *implies* $x\sigma \neq y\sigma$. *Then* $\pi(\mathrm{H}(\sigma))$ *is conservative.*

Proof idea: As $x\,\pi(\mathrm{H}(\sigma)) = \pi(H(x\sigma))$*, we consider all possible (by definition) terms* $s \in H(\cdot)$. *Remark that for case* $s = \mathrm{bin}(\pi(H(p)), \pi(H(q)))$*, it is always possible to choose* $u \in \mathrm{QSub}(\mathcal{S})$ *such that* $\ulcorner u\sigma \urcorner = \ulcorner \mathrm{bin}(t_1,t_2) \urcorner$ *and* $\sqrt{u} \neq$ "$\cdot$"*. Then using Proposition 3 we can show that* $s = u\,\pi(\mathrm{H}(\sigma))$. ■

C. Bounds on conservative solutions

To get a decidability result, we first show an upper bound on size of conservative model and then, by reducing any satisfiable constraint system to one that have conservative model and showing that reduced one is smaller (by size) than original one, we obtain an existence of a model with bounded size for any satisfiable constraint system.

Lemma 4. *Given a normalized constraint system $\mathcal{S}$ and its conservative model σ. Then* $\forall x \in \mathrm{Vars}(\mathcal{S}), \mathrm{QSub}(x\sigma) \subseteq \ulcorner \mathrm{QSub}(\mathcal{S})\sigma \urcorner \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})$.

Proof idea: Let $\sqsubset$ be the strict total order on $\mathrm{Vars}(\mathcal{S})$ such that $\mathrm{size}(x\sigma) < \mathrm{size}(y\sigma)$ implies $x \sqsubset y$. Then we can show that a set of quasi-subterms of $\mathcal{S}$ whose σ-instances are in $\mathrm{elems}(x\sigma)$ consists only of terms that do not contain bigger (w.r.t. $\sqsubset$) variables than x. Finally, by induction on the well ordered set $(\mathrm{Vars}(\mathcal{S}), \sqsubset)$ we can prove the lemma. ■

Proposition 5. *For normalized constraint system $\mathcal{S}$ that have a conservative model σ,* $\forall x \in \mathrm{Vars}(\mathcal{S}), \mathrm{size}(x\sigma) \leq 2 \times \mathrm{size}(\mathcal{S})$.

Proof: From Lemma 4, $\mathrm{size}(x\sigma) = |\mathrm{QSub}(x\sigma)| \leq |\ulcorner \mathrm{QSub}(\mathcal{S})\sigma \urcorner \cup \mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})| \leq |\ulcorner \mathrm{QSub}(\mathcal{S})\sigma \urcorner| + |\mathrm{priv}(\mathrm{QSub}(\mathcal{S}) \cap \mathcal{A})| \leq \mathrm{size}(\mathcal{S}) + \mathrm{size}(\mathcal{S})$. ■

From this proposition and Proposition 4 we obtain an existence of bounded model for a normalized constraint system that have a model sending different variables to different values. We will reduce an arbitrary constraint system to the already studied case.

Theorem 2. *Constraint system $\mathcal{S}$ is satisfiable iff then there exists a model σ of $\mathcal{S}$ such that* $\forall x \in \mathrm{dom}(\sigma), \mathrm{size}(x\sigma) \leq 2 \times \mathrm{size}(\ulcorner \mathcal{S} \urcorner)$.

Proof idea: One direction is trivial. For the other, given a normalized model σ' of $\mathcal{S}$ we build a substitution θ that maps different variables σ'-instances of those is the same to one variable. Thus we can apply Proposition 4 on $\ulcorner \mathcal{S}\theta \urcorner$ and get its conservative model σ''. By applying Proposition 5 we get a bound on size for this model. By showing that $\mathrm{size}(\ulcorner \mathcal{S}\theta \urcorner) \leq \mathrm{size}(\ulcorner \mathcal{S} \urcorner)$ and that σ'' applied to θ (it has the same values as σ'' on its domain) is a model of $\mathcal{S}$, we can conclude. ■

Thus, we propose simple steps of non-deterministic algorithm to decide the satisfiability of constraint system $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$:

1) Guess for every variable $x \in \mathrm{Vars}(\mathcal{S})$ value of a ground substitution σ such that $\mathrm{size}(x\sigma) \leq 2 \times \mathrm{size}(\mathcal{S})$.
2) Check for all i whether $\ulcorner t_i \urcorner \in \mathrm{Der}(\ulcorner E_i\sigma \urcorner)$:
   a) if the check is positive for some i, then $\mathcal{S}$ is satisfiable with σ;
   b) if the check is negative for all i, then $\mathcal{S}$ is not satisfiable.

The correctness of this algorithm follows from Theorem 2.

IV. Complexity analysis

In this section we give the complexity of used algorithms. First, we define a measure for inputs. For the size of terms and set of terms, we take $\mathrm{size}(\cdot)$. For system of constraints $\mathcal{S} = \{E_i \rhd t_i\}_{i=1,\ldots,n}$ we take $n \times \mathrm{size}(\mathcal{S})$.

Proposition 6. *Checking $t \in \mathrm{Der}(E)$ is polynomial in $\mathrm{size}(E \cup \{t\})$.*

Proof idea: To build t from E, we show it is sufficient to consider the quasi-subterms of $E \cup \{t\}$; then we iteratively add to a set of terms S (initially equal to E) all the terms from $\mathrm{QSub}(E \cup \{t\})$ that can be deduced in one step. When a fix point is reached, we have $t \in S$ iff $t \in \mathrm{Der}(E)$. ■

Lemma 5. *Computing $\ulcorner t \urcorner$ is polynomial in $\mathrm{size}(t)$.*

Proof idea: The algorithm of normalization works bottom-up by flattening nested ACI-sets, sorting the children of ACI operator, removing duplicates and removing nodes without incoming edges (except the root node of t). ■

Proposition 7. *Satisfiability of general DY+ACI constraint systems is in NP.*

Proof idea: We have to show that the verification of a model is polynomial with regard to the initial problem size. To do this, we normalize $\mathcal{S}\sigma$ and then apply the algorithm for checking ground derivability. Using the fact that $\mathrm{size}(x\sigma) \leq 2 \times \mathrm{size}(\mathcal{S})$, Lemma 5 and Proposition 6, we can bound the execution time with a polynomial on $n \times \mathrm{size}(\mathcal{S})$. ■

On the other hand, we can reuse a technique presented in [13] to show that the satisfiability of a constraint system is an NP-hard problem. The authors encoded 3-SAT problem into an insecurity problem of a single-session sequential protocol. Because the steps of the protocol are linearly ordered, the finding of an attack is reduced to the satisfiability problem of a single constraint system.

Theorem 3. *Satisfiability of general DY+ACI constraint systems is NP-complete.*

V. Satisfiability of general DY constraints

The previous result on constraint solving for DY+ACI theory can be projected to the classical DY case. We cannot apply it directly, as in the resulting model we probably meet an ACI symbol. First, we can show that if a constraint system is satisfiable within DY, then it is satisfiable within DY+ACI. Second, we find a model of a given constraint system within DY+ACI. And third, we will transform the model within DY+ACI in such a way that the resulting substitution will be a model of initial constraint system within DY. The idea of satisfactory transformation δ is simple: we replace any ACI list of terms with nested pairs: $\cdot(t_1, \ldots, t_n)$ we replace with $\mathrm{pair}(t_1, \mathrm{pair}(\ldots, t_n))$. Note that this transformation will have a linear complexity and the transformed model will have the DAG-size less than double initial size. This gives us an NP procedure for the problem of general constraint systems satisfiability within DY model. Theorem **??** holds for the DY constraint systems.

Example 6. *Let us consider a standard constraint system similar to one in Example 2.*

$$\mathcal{S} = \left\{ \begin{array}{lcl} \mathrm{enc}(x, a), \mathrm{pair}(c, a) & \rhd & b \\ \mathrm{pair}(x, c) & \rhd & a \end{array} \right\},$$

We know how to get a model of $\mathcal{S}$ within DY+ACI; let it be one as in Example 5: $\sigma = \{x \mapsto \cdot(a, b, c)\}$.

Then, by applying transformation $\delta(\cdot)$, we will get $\sigma' = \delta(\sigma) = \{x \mapsto \text{pair}(a, \text{pair}(b, c))\}$. We can see that σ' is also a model of $\mathcal{S}$ within DY.

VI. Conclusions

In this work we have presented a decision algorithm for solving Dolev Yao general constraints as well as ones extended with an ACI symbol, which can be used to represent sets of terms. The complexity of the algorithm was proved to be in NP.

We have given also two applications of the presented result: protocol insecurity with non-communicating intruders and discovering XML-based attacks.

Further work is aimed to extend the results to other deduction systems and equality theories.

References

[1] J. Millen and V. Shmatikov, "Constraint solving for bounded-process cryptographic protocol analysis," in *Proceedings of the 8th ACM conference on Computer and Communications Security*, ser. CCS '01. New York, NY, USA: ACM, 2001, pp. 166–175.

[2] D. Basin, S. Mödersheim, and L. Viganò, "Ofmc: A symbolic model checker for security protocols," *International Journal of Information Security*, vol. 4, pp. 181–208, Jun. 2005.

[3] M. Turuani, "The CL-Atse Protocol Analyser," in *Term Rewriting and Applications (RTA)*, 2006, pp. 277–286.

[4] C. Cremers, "The Scyther Tool: Verification, falsification, and analysis of security protocols," in *Computer Aided Verification, 20th International Conference, CAV 2008, Princeton, USA, Proc.*, ser. Lecture Notes in Computer Science, vol. 5123/2008. Springer, 2008, pp. 414–418.

[5] D. A. Basin, S. Mödersheim, and L. Viganò, "Algebraic intruder deductions," in *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, 2005, pp. 549–564.

[6] Y. Chevalier and M. Rusinowitch, "Symbolic protocol analysis in the union of disjoint intruder theories: Combining decision procedures," *Theoretical Computer Science*, vol. 411, no. 10, pp. 1261 – 1282, 2010, iCALP 2005 - Track C: Security and Cryptography Foundations.

[7] V. Cortier, S. Delaune, and P. Lafourcade, "A survey of algebraic properties used in cryptographic protocols," *Journal of Computer Security*, vol. 14, no. 1, pp. 1–43, 2006.

[8] S. Delaune, P. Lafourcade, D. Lugiez, and R. Treinen, "Symbolic protocol analysis for monoidal equational theories," *Information and Computation*, vol. 206, no. 2-4, pp. 312–351, Feb.-Apr. 2008.

[9] L. Mazaré, "Satisfiability of Dolev-Yao Constraints," *Electronic Notes in Theoretical Computer Science*, vol. 125, no. 1, pp. 109–124, 2005.

[10] L. Mazaré, "Computational Soundness of Symbolic Models for Cryptographic Protocols," Ph.D. dissertation, Institut National Polytechnique de Grenoble, Oct. 2006.

[11] P. Syverson, C. Meadows, and I. Cervesato, "Dolev-Yao is no better than Machiavelli," in *First Workshop on Issues in the Theory of Security — WITS'00*, 2000, pp. 87–92.

[12] M. Baudet, "Deciding security of protocols against off-line guessing attacks," in *ACM Conference on Computer and Communications Security*, 2005, pp. 16–25.

[13] M. Rusinowitch and M. Turuani, "Protocol insecurity with a finite number of sessions, composed keys is np-complete," *Theor. Comput. Sci.*, vol. 1-3, no. 299, pp. 451–475, 2003.

[14] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani, "Deciding the security of protocols with Diffie-Hellman exponentiation and products in exponents," in *FSTTCS*, 2003, pp. 124–135.

[15] ——, "An np decision procedure for protocol insecurity with xor," *Theor. Comput. Sci.*, vol. 338, no. 1-3, pp. 247–274, 2005.

[16] V. Shmatikov, "Decidable analysis of cryptographic protocols with products and modular exponentiation," in *In Proc. 13th European Symposium on Programming (ESOP '04), volume 2986 of LNCS*. Springer-Verlag, 2004, pp. 355–369.

[17] S. Narayanan and S. McIlraith, "Simulation, verification and automated composition of web services," in *Proceedings of the Eleventh International World Wide Web Conference (WWW-11)*, Honolulu, Hawaii, USA, May 7-11 2002, pp. 77–88.

[18] V. Cortier, B. Warinschi, and E. Zalinescu, "Synthesizing secure protocols," in *Computer Security - ESORICS 2007, 12th European Symposium*. Springer, 2007, pp. 406–421.

[19] Y. Chevalier, D. Lugiez, and M. Rusinowitch, "Towards an automatic analysis of web service security," in *FroCoS 2007, Liverpool, UK, September 10-12*, ser. Lecture Notes in Computer Science, vol. 4720. Springer, 2007, pp. 133–147.

[20] OWASP Foundation, "OWASP-DV-008, OWASP Testing Guide, v3.0," 2008. [Online]. Available: http://www.owasp.org/index.php/Testing_for_XML_Injection_(OWASP-DV-008)

[21] T. Avanesov, Y. Chevalier, M. Rusinowitch, and M. Turuani, "Satisfiability of General Intruder Constraints with and without a Set Constructor," INRIA, Research Report RR-7276, 05 2010. [Online]. Available: http://hal.inria.fr/inria-00480632/en/